FILESYSTEMS

NFS-mounted home directories (~3 GB limit) and project spaces (**/usr/projects**) on all front ends, master nodes, and slave nodes. Project spaces are auto-mounted, so **ls** may not "see" them until after you **cd** into them.

Make sure your home space does not exceed quota because if it does you won't be able to **llogin**.

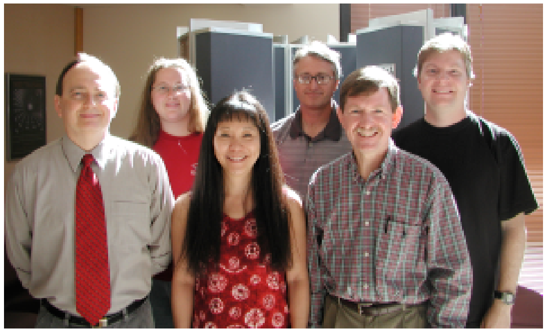
30 TB Panasas global temporary scratch space (no per-user limits) **/net/scratch1** and **/net/scratch2** on all front ends, master nodes, and slave nodes, not visible to other LANL clusters. You may have to **mkdir *moniker*** to use these spaces the first time. http://computing.lanl.gov/article/439.html has details on I/O optimization.

PSI COMMANDS

chacl	{[**-clear** | **-f** *fname* | **-rm** *entry* | **-update** *entry*]} [**-d** *dir*] [**-Q**] [**-R**] *filelist*
chgrp	**-d** [*dir*] [**-Q**] [**-R**] *grp filelist*
cp	[**-cond**] [**-d** *dir*] [**-max** *n*] [**-min** *n*] [**-Q**] [**-showBlkRate**] [**-showConfig**] [**-showRate**] [**-simFiles** *n*] [**-t**] [**-tape**] *file1 file2*
get	[**-cond**] [**-d** *dir*] [**-doff** *n*] [**-len** *n*] [**-max** *n*] [**-min** *n*] [**-norestore**] [**-passive**] [**-Q**] [**-R**] [**-showBlkRate**] [**-showConfig**] [**-showRate**] [**-simFiles** *n*] [**-soff** *n*] *filelist*
ls	[**-1**] [**-A**] [**-a**] [**-C**] [**-d**] [**-F**] [**-g**] [**-h**] [**-I**] [**-k**] [**-l**] [**-M**] [**-P**] [**-r**] [**-R**] [**-s**] [**-S**] [**-t**] [**-V**] [*filelist*]
lsacl	*fname*
mkdir	[**-cmt** *comm*] [**-cond**] [[**-d** *dir*] [**-p**] [**-Q**] *dirlist*
mv	[**-d** *dir*] [**-Q**] [**-t**] *file1 file2*
quit	
rm	[**-d** *dir*] [**-i**] [**-Q**] [**-r**] [**-R**] [**-t**] *filelist*
rmdir	[**-d** *dir*] [**-Q**] *dirlist*
save	(See **store** command)
status	
store	[**-cmt** *comm*] [**-cond**] [**-d** *dir*] [**-doff** *n*] [**-len** *n*] [**-max** *n*] [**-min** *n*] [**-passive**] [**-Q**] [**-R**] [**-rm**] [**-serial**] [**-showBlkRate**] [**-showConfig**] [**-showRate**] [**-simFiles** *n*] [**-soff** *n*] [**-t**] [**-tape**] [**-vault**] *filelist*
undelete	[**-d** *dir*] *filelist*

LA-UR-04-8933 issued January, 2005. Last updated (hjw) 04/21/06.

Steven R. Shaw
High Performance Computing Systems
(CCN-7) Group Leader
505-606-0203



Left to right: Roger Martz, Meghan Quist, Sara Hoshizaki, Hal Marshall, David Kratzer, Jeff Johnson. Not pictured: Robert Cunningham, Robert Derrick.

Contact ICN Consulting for any questions and issues related to Flash.

ICN CONSULTING

consult@lanl.gov

505-667-5745

http://computing.lanl.gov

Los Alamos
NATIONAL LABORATORY
EST. 1943

QUICK REFERENCE CARD
FLASH

Complete documentation available on http://computing.lanl.gov

OVERVIEW

A protected (yellow) network supercomputer cluster with dual-processor, 1-MB L2 cache, AMD Opteron nodes and Myrinet interconnect. Operating system is Linux + BProc (a kernel modification that allows parts of one node's process space to exist on other nodes, even if those nodes are running their own private Linux kernel). 1,906 total compute nodes but is managed as 5 individual segments and user jobs cannot span segments. Four segments (flasha, flashb, flashc, and flashdev) currently operate in 32-bit (Opteron "legacy") mode (maximum 2-GB **malloc**); the fourth (flashd) is in 64-bit mode. Each segment has one BProc master node that does not run production jobs and some number of BProc compute nodes that do (flasha=300; flashb=flashc=255; flashd=127; flashdev=15). Flasha CPUs are 2.0 GHz; all others are 2.4 GHz. 8 GB memory per node, except flashd (16 GB).

LOGGING IN

Three front-end/ssh gateways: ffe1 and ffe2 (32-bit) and ffe-64 (64-bit) from which you can access the entire cluster. Use **ssh** to a front end and authenticate with CryptoCard.

COMPILING / PREPARING to RUN

All compiling/linking must be done on the front-end systems (ffe1, ffe2, and ffe-64).
DO NOT **`llogin`** before compiling.

All system software (compilers, tools, debuggers, MPI) must be accessed through the **`module`** utility before both compiling and running. Modulefiles on BProc systems are of the form package/version; e.g., pgi/5.1 or lampi/1.5.12. Most packages have a default version that can be used without specifying the version.

List all available modulefiles:
```
 module avail
```
List modulefiles currently loaded:
```
 module list
```
Add a modulefile to current environment:
```
 module load modulefile
```
Remove a modulefile from current environment:
```
 module unload modulefile
```
Replace modulefile1 with modulefile2:
```
 module switch modfile1 modfile2
```

32-bit compilers are:
GNU (**g77|gcc|g++**); Lahey (**lf95**);
Intel7.1 (**ifc|icc**); Intel 8 (**ifort|icc**);
Absoft (**f77|f90|f95**); Portland Group
(**pgf77|pgf90|pgcc**); NAG (**f95**)

64-bit compilers will be:
GNU (**g77|gcc|g++**); Portland Group
(**pgf77|pgf90|pgcc**); PathScale
(**pathf90|pathcc|pathCC**)

There are no shells on the slave nodes. Any shell-script commands execute on the master or front-end nodes. There is no perl on the slave nodes, although there is perl emulation within BProc (see **`man BProc`**).

MPI available via LAMPI or OpenMPI:
```
module load lampi/version
```

For both Fortran and C: include **`mpi.h`**, link with **`–lmpi`**, and add the following two compile/link flags, e.g.,:
```
f90 file.f $MPI_COMPILE_FLAGS
$MPI_LD_FLAGS -lmpi
```

RUNNING JOBS with LSF (Load Sharing Facility)

There will be 4 Flash LSF execution hosts. flasha, flashb, and flashc run 32-bit production; flashd will run 64-bit.

For interactive use, first obtain an allocation of slave nodes with **`llogin [-n #]`** . Result is an interactive shell on a BProc master node (*ll*-1 – *ll*-6 and *lb*-1 – *lb*-7) and an allocation of ***`#/2`*** slave nodes. Default is one node. Two environment variables are set by LSF: **`$NODES`** (list of nodes allocated) and **`$NODELIST`** (list of processors allocated). See **`man llogin`**.

Run a serial interactive job (after **`llogin`**):
```
bpsh $NODES ./a.out.serial
```

Submit a serial batch job:
```
bsub [bsub options] 'bpsh $NODES
./a.out.serial'
```

Run a parallel interactive job (after **`llogin –n #`** and **`module load mpi/mpi-version`**):
```
mpirun –np # ./a.out.MPI
```

`bsub` options (just a few; see **`man bsub`**):
```
 [-L /bin/tcsh][-q queue_name]
  [-o out_file] [-e err_file]
  [-n #procs] [-W [hours:]minutes]
  [-m host] -wa URG -wt 20 …
```

Using **`bsub`** job scripts:
```
 bsub < scriptname
```
where **`scriptname`** contains:
```
 #! /bin/tcsh
 #BSUB -q queuename
 #BSUB -o output_file
 mpirun –np …
```

Debug serial job (after **`llogin`**):
```
module load totalview
totalview -remote $NODES ./a.out
```

Debug parallel job (after **`llogin –n #`**):
```
module load debugger/tv-version
totalview mpirun –a –np # ./a.out
```

Submit a job to a 64-bit segment via **`bsub (`**or **`llogin) –q flash64q`**

MONITORING JOBS

LSF job information for all segments and front ends:
```
  bjobs [-l] [-u user] [JobID#]
```

System information for only the segment on which the command is executed:

`bpstat` shows which users are assigned which nodes via LSF.
`bpps [-n] [-u user] [-l] [-s]` displays current BProc process status.
`ps –elf` reports status for all processes but slave node processes are shown in [square brackets]
`bpsh $NODES ps axmv` shows dynamic memory usage on slave node for a running job.
`top` displays ongoing look at processor activity (slave node processes *not* shown in square brackets).
`bptop` displays ongoing look at processor activity; type '**`c`**' to toggle between master-node and slave-node processes.

UTILITIES and TOOLS

Give (copy) a file to a user:
```
  give filename userid
```
(result is in
```
  /net/scratch1/givedir/userid)
```

Processor admin information:
```
  cat /proc/cpuinfo
  cat /proc/meminfo
```

Show or change process limits for current shell:
```
  limit/unlimit
```

HPSS

No PSI access from slave nodes. Submit large HPSS transfer jobs to LSF queue *ftaq*.

LINKS

http://computing.lanl.gov/	main documentation site
http://icnn.lanl.gov/	status, news
http://icnn.lanl.gov/drm/alljobs	job monitoring
http://asci-training.lanl.gov/	training materials
http://storage.lanl.gov/	storage information